## REMARKS/ARGUMENTS

Claims 1-33 are pending in this application, all of which stand rejected. Claims 1, 3-6, 9, 11, 13-21, 24, 26-30, 32, and 33 have been rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,845,281 (Benson). Claims 2, 7, 8, 10, 12, 22, 23, 25, and 31 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Benson in view of U.S. Patent No. 5,708,709 (Rose). Following entry of the present amendment, claims 1, 8, and 9 will have been amended.

For the reasons set forth below, applicants respectfully submit that the claims, as amended, are patentable over the references cited and that this case is now in condition for allowance.

Independent claims 1, 9, 16, and 28

The independent claims of the present application (claims 1, 9, 16, and 28) have been rejected under section 102(b) as being anticipated by Benson. For the reasons set forth below, the independent claims of the present application define features that are not taught or suggested by Benson.

Benson describes a system that manages the use of data objects in such a way that usage of the data objects complies with a set of conditions. As described in Benson, an author creates a data object, and also determines the conditions for usage of the data object. (Benson, col. 5, ll. 16-17, 23-24.) The data objects and usage conditions are provided to a "packaging program," which creates a "secure data package" containing the data object and the usage conditions. (Benson, col. 5, ll. 24-28.) Once the data object has been packaged by the packaging program, the data object can only be accessed by a "user program." (Benson, col. 5, ll. 28-29.) After the data object has been packaged, the package is received by a user. (Benson, col. 5, ll. 60-61.) The user must use the "user program" to unpackage the data object from its package and obtain the data in a usable form. (See Benson, col. 5, ll. 61-63.) After the data object has been used by the user, the data is repackaged back into a secure package. (Benson, col. 5, ll. 63-64.)

In contrast with Benson, each of the independent claims defines one or more features that is not taught or suggested by Benson. The following describes how each of the independent claims defines over Benson.

*Claim 1*

Claim 1 defines both a "rendering application" and a "management module." The rendering application communicates with the management module to access stored encrypted information, and the management module returns data that enables the rendering application to use the information. Benson does not teach or suggest this feature.

Benson shows a single, self-contained "user program" (see FIG. 14) that receives, decrypts, and renders a data package, without assistance from any separate components. Claim 1, by contrast, recites both a rendering application and a management module. The rendering application and management module communicate with each other: the rendering application communicates with the management module in order to access encrypted information, and the management module responds with data that enables the rendering application to use the information. No such communication between two components is described in Benson, since Benson's "user program" contains all of the capability it needs to render, and enable use of, decrypted content.

Applicants have amended claim 1 to more particularly point out the nature of, and relationship between, the rendering application and the management module. In particular, claim 1, as amended, recites that the management module is separate from the rendering application, and that the management module is interfaceable to any one of a plurality of rendering applications. Thus, the components of claim 1 are modularizable and reusable in a way that the single, unitary "user program" of Benson is not.

It should be noted that, in the Office Action, the Examiner has not addressed the fact that the rendering application communicates with a management module. Rather, the Examiner has merely stated that Benson's "usage manager module" (which is shown in FIG. 14 as a component of the "user program") corresponds to the claimed "rendering application," and that the "usage management module" is used to access data. In other words, Benson's structure uses one component (the "user management module") to perform all functions, while the invention recited in claim 1 uses a separate rendering application and

management module which communicate with each other. This fact represents a structural difference between claim 1 and Benson. In view of this structural difference, Benson does not anticipate claim 1.

*Claim 9*

Claim 9 calls for the act of "authenticating software," and "providing at least one cryptographic service for said software." Additionally, claim 9, as amended, defines the cryptographic service as being performable by computer-executable instructions that are separate from the software and invocable by a call from the software. This structure is not taught or suggested by Benson, for at least two reasons.

First, with regard to the authenticating feature, Benson does not teach the act of authenticating *software*. At best, Benson describes the authentication of a *user* who will use the software. The American Heritage Dictionary (4th ed. 2000) defines "authenticate" to mean "To establish the authenticity of; prove genuine." The Examiner reads the authentication feature onto col. 11, ll. 66-67 of Benson, which states that use of Benson's "user program" may be controlled by a password. At best, the password procedure described in Benson proves the identity of the user, but does not in any way establish that the software is authentic or genuine. In Benson's password procedure, the password may prove that the user is who he or she claims to be, while the software may be a complete forgery, or may have been tampered with. Thus, Benson's password procedure does not teach the claimed feature of "authenticating software."

Second, claim 9, as amended, defines computer-executable instructions that (1) are separate from the software that has been authenticated, and (2) provide a cryptographic service for that software. In other words, the instructions recited in claim 9 perform a cryptographic service for a separate piece of software, but only after the authenticity of the receiving software has been verified. Thus, the invention recited in claim 9 allows the process of cryptography to be modularized, while reducing the risk that valuable cryptographic services will be provided for a counterfeit program. Even if Benson's password procedure could be considered to "authenticate" the user program, there is no set of instructions that is separate from the user program and provides a cryptographic services for the user program, since Benson's user program contains a decryption module (element 1405, FIG. 14) that

performs decryption from inside the user program itself. Thus, Benson does not teach or suggest the features of claim 9, as amended.

*Claim 16*

Claim 16 recites the acts of providing two different components to two different entities. A first entity is provided with an interface that is used to request a service, and a second entity is provided with a set of computer-executable instructions that provide the service and that are invocable by way of the interface. Benson does not teach the provision of these components to two separate entities.

As described above, Benson teaches a single "user program," that includes everything needed to use a data package. There is no teaching in Benson that different parts of the user program are provided to different entities. Rather, the entire user program exists at the user's computer. In paragraph 4 of the Office Action, the Examiner indicates that the user program, and the usage manager module contained in the user program, are separate entities. This reasoning is incorrect for two reasons. First, the user program (and its contained "usage manager module") is simply a computer program that resides on a computer; there is no interface or set of instructions that is "provided" to the user program (or to the "usage manager module" that is part of the computer program). Second, neither the program, or the manager module, is an entity in the sense of claim 16. As described in the present application: "In one example, the developer/administrator of architecture 90' may provide a specification or description of interface (e.g., a set of method names/labels for the API) to the developer of the reader 92, and may then provide a DLL or COM object (or successive DLLs and COM objects) to the users of client architecture 90'." (Application, p. 11, lines 13-16.) Thus, the present application contemplates that the interface, and the instructions invoked by the interface, can be provided to two separate people and/or businesses, which are examples of the kinds of "entities" recited in claim 16. There is no reasonable construction of the term "entity" (either based on the specification, or any dictionary definition) under which a program, and one the components included inside that program, can be considered separate "entities."

Thus, Benson does not teach the features of claim 16.

*Claim 28*

Claim 28 calls for issuing a request to provide data, conditional upon "first information" being sealed with "information pertaining to an authorized user of said first information." Benson contains no reference to sealed data. An electronic search of the text of Benson reveals that Benson does not contain the word "seal" or "sealed." Moreover, while the user of Benson's "user program" arguably issues a request for data (i.e., by requesting to use the content contained in a secure data package), there is no teaching or suggestion in Benson that the issuance of such a request is conditional upon whether one piece of information is sealed (or associated in any other way) with "information pertaining to an authorized user of" the information.

Thus, Benson does not teach or suggest the features of claim 28.

Dependent claims

Since the independent claims have been demonstrated to be patentable, the dependent claims are patentable at least by reason of their dependency. Additionally, certain dependent claims are patentable for reasons that are cumulative to the patentability of the independent claims on which they depend. The following is a non-exhaustive list of dependent claims that are patentable for separate reasons:

*Claim 6*

Claim 6 (which is dependent on claim 1) recites that the management module authenticates the rendering application. As discussed above in connection with claim 16, Benson does not teach that one component authenticates another. In particular, the Examiner has cited (1) col. 9, ll. 55-56, and (2) col. 12, ll. 50-57 of Benson as teaching this authentication. (See Office Action, ¶ 7.) Cited portion (1) discusses the transmission of credit card information; cited portion (2) discusses whether a usage request is permitted by control data. Neither of these cited portions has anything to do with authenticating a rendering application – i.e., evaluating whether a program is authentic.

Thus, for this additional reason claim 6 is patentable over the cited references.

*Claims 17 and 19*

Claims 17 and 19 are dependent on claim 16. As noted above, claim 16 calls for an interface and a set of instructions to be provided to two different entities. As further noted above, examples of such entities are businesses or people. Claim 17 recites that the first entity is a software developer, and claim 19 recites that the second entity is a consumer of information. Benson does not teach or suggest the provision of an interface and a set of instructions to these two different entities. The Examiner's argument is that first entity is Benson's user program and the second entity is a module contained in the user program; this reading is clearly incorrect.

Thus, claims 17 and 19 are patentable over the prior art of record for these additional reasons.

*Claim 27*

Claim 27 calls a secure repository to be invoked if there is one level of protection for a piece of information, and for use of the information to be enabled without a secure repository if there is another level of protection. Benson does not teach these features. Benson makes no reference to secure repositories, levels of protection, or to the feature of allowing used of the data to be enabled with or without a secure repository depending on the level of protection. The cited portion of Benson (col. 12, ll. 50-60) discusses that use of content may be either allowed or not allowed, depending on the control data associated with the content. The features recited in claim 27 are not directed to deciding whether use of the content should be allowed or not allowed; they are directed to deciding whether use of the content should be allowed with, or without, a secure repository.

Thus, Benson does not teach or suggest the features of claim 27.

*Claim 7*

Claim 7 has been rejected under section 103(a) over a combination of Benson and Rose. Claim 7 defines a repository that is receivable by way of a network infrastructure, where the repository applies a key. Rose makes no mention of repositories, or the receipt of repositories by way of a network infrastructure. Nor does anything in Rose suggest such a repository, or its receipt over a network infrastructure. The cited portion of Rose (col. 5, ll.

31-49) describe general cryptographic processes, but make no mention of repositories, or their receipt through a network. Nor does Benson suggest repositories, or their receipt through a network.

Thus, the proposed combination of Benson and Rose does not make claim 7 obvious.

*Claim 12*

Claim 12 recites sealed data that comprises a cryptographic key. Neither Rose nor Benson teaches or suggests the use of sealed data. As noted above, a digital scan of Benson reveals that the terms "seal" and "sealed" do not appear in Benson. Similarly, those terms do not appear in Rose. Nothing in Benson or Rose teaches or suggests the sealing of keys (or any other data). The Examiner has cited col. 5, ll. 31-49 of Rose as teaching the features of claim 12. This cited portion refers only to cryptographic processes in general, but does not mention, or suggest, sealing.

Thus, the proposed combination of Benson and Rose does not make claim 7 obvoius.

Explanation of claim amendments

Claim 8 has been amended to correct a minor typographical oversight in its dependency. The amendment to claim 8 is not made for a reason related to patentability, and is not intended to alter the scope of claim 8.

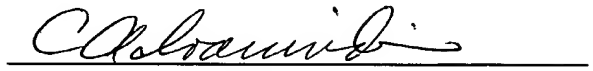
Claim 9 has been amended to correct a minor typographical oversight. Specifically, the reference to "first software" has been changed to "software," to make that reference consistent with the term's prior usage in the claim. This elimination of the term "first" is not made for a reason related to patentability, and is not intended to alter the scope of claim 9.

Moreover, no new matter has been added by the amendments to claims 1, 8, and 9. Claim 8 merely changes the numeral of the dependent claim to which it refers. Claims 1 and 9 have been amended to recite the separateness of two software modules, and this separateness is shown in the application at least at FIG. 3 and pages 10-11.

Conclusion

For all of the foregoing reasons, applicants respectfully submit that the claims are patentable over the prior art of record and are in condition for allowance.

Date: March 23, 2004



Christos A. Ioannidi
Registration No. 54,195

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439